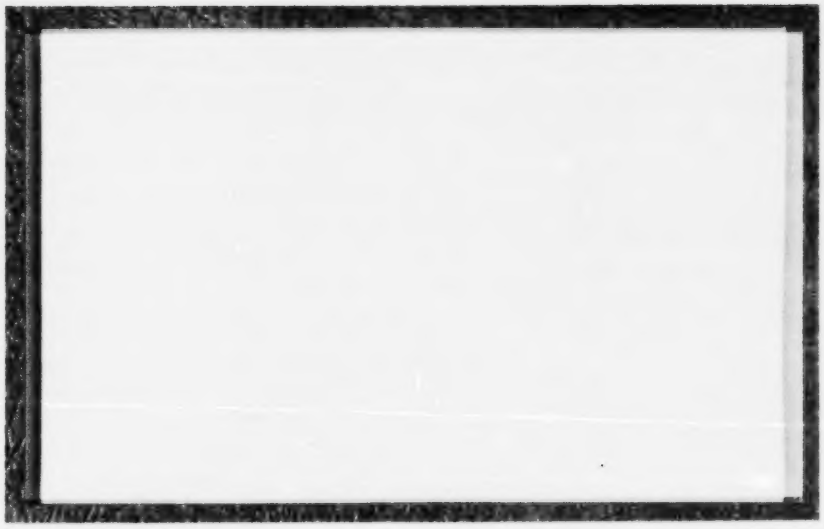
UNIVERSITÉ DE MONTRÉAL



CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC H3C 3J7

EXACT ALGORITHMS FOR THE VEHICLE ROUTING PROBLEM

by

Gilbert Laporte*
Yves Nobert**

* École des Hautes Études Commerciales de Montréal

** Département des Sciences administratives, Université du Québec à Montréal.

The authors are grateful to the Canadian Natural Sciences and Engineering Research Council (grants A4747 and A5486) and to the Quebec Government (FCAC grant 80EQ0428) for their financial support.

Cette étude a été publiée grâce à une subvention du fonds F.C.A.R. pour l'aide et le soutien à la recherche.

Université de Montréal

Centre de recherche sur les transports - Publication #480
École des Hautes Études Commerciales - Cahier du GERAD #G-85-11

June 1985 - revised in November 1985

Dépôt légal - 3e trimestre 1986

ABSTRACT

This paper presents a survey of exact algorithms for various types of vehicle routing problems. Algorithms are divided into three main categories: direct search methods, dynamic programming and integer linear programming. This latter category comprises three approaches: set partitioning, vehicle flow formulations and commodity flow formulations.

RESUME

On présente dans cet article une revue des algorithmes exacts pour plusieurs types de problèmes de tournées. Les algorithmes sont répartis en trois catégories: méthodes de recherche directe, programmation dynamique et programmation linéaire en nombres entiers. Cette dernière catégorie inclut trois approches: recouvrement d'ensembles, flots de véhicules et flots de commodités.

1. INTRODUCTION

In the broadest sense, the vehicle routing problem (VRP) can be described as the problem of designing optimal delivery routes from one or more depots to a set of geographically scatterred points, cities or customers. As noted elsewhere [8], the term delivery can sometimes be replaced by collection and there exist situations not involving the actual transportation of goods. Practical applications of the VRP are numerous and encompass many spheres of activity. Their economic importance cannot be overstated. In dollar terms alone, Fisher and Jaikumar [25] estimated that in 1975 in the United States, delivery vehicles consumed approximately $5.7 billion worth of fuel. This says nothing of other costs involved in physical distribution. Thus, any saving generated by improvements in route design methodology may be significant.

The VRP has received the attention of many Operations researchers over the last three decades. This interest is due in part to the practical importance of the problem, but also to its intrinsic difficulty. Only recently, the largest problems of any complexity which had been solved by exact algorithms contained approximately 30 points [4]. This figure has since been improved, but only in particular instances of the problem [18,21,27,49,50,55]. The VRP constitutes a generalization of the travelling salesman problem (TSP) which consists of determining the shortest circuit or cycle passing through each of n points once and only once [14]. The TSP and the VRP are both NP-complete [58]. However, from a practical point of view, VRPs are in general much more difficult to solve than TSPs of the same size. This renders their study all the more challenging. Also, while much effort has been devoted to the solution of the TSP by exact methods, most algorithms for the VRP are heuristics, reflecting the relative difficulty of the problem. Promising avenues of research have been opened but many are still unexplored [60].

The object of this paper is to present a survey of *exact* algorithms for the VRP, while emphasizing recent results and avoiding

as far as possible, duplication of previous work. Several surveys and bibliographies have indeed been published on the VRP (see for example [3,4,8,24,38,52,58,60,67,70,75,78]), most of which have tended to concentrate on heuristic methods. The state-of-the-art paper of Bodin et al. [4] is fairly exhaustive in this respect. Also of major interest are the typology proposed by Bodin and Golden [3] and the analysis of research perspectives and prospects presented by Magnanti [60]. We adopt, whenever possible, the definitions and classifications of these authors.

1.1 Notation and definitions

Some notation and a more formal definition of the VRP are required at this point. Let $G = (N,E,C)$ be a graph where $N = \{1,\ldots,n\}$ is a set of nodes representing cities or customers, E is a set of arcs or edges (undirected arcs) and $C = (c_{ij})$ is a distance matrix associated with E. C is symmetrical if and only if $c_{ij} = c_{ji}$ for all $i,j \in N$. It satisfies the triangle inequality if and only if $c_{ij} + c_{jk} \geq c_{ik}$ for all $i,j,k \in N$. Problems fall into two broad categories: symmetrical problems (those for which C is symmetrical) and asymmetrical problems. Algorithms which exploit this distinction generally require that E be defined as a set of edges in the symmetrical case and as a set of arcs in the asymmetrical case. Also, depending on the type of algorithm c_{ii} may be equal to 0, to infinity or simply left undefined. Let $R = \{1, \ldots, r\}$ $(r < n)$ be a set of depots. It is assumed that a fleet of m vehicles have based at the depots. The value of m is sometimes fixed a priori (it is then set equal to $\bar{m}$) or may constitute a decision variable. In general, one can impose bounds on m:

$$(1) \quad \underline{m} \leq m \leq \bar{m}$$

Similarly, there may be bounds on the number of vehicles assigned to each depot. Fixed costs may be imposed on the use of a depot or of a vehicle.

The VRP consists of establishing minimum cost vehicle routes in such a way that

(i) each city in N-R is visited exactly once by exactly one vehicle;

(ii) each vehicle starts and ends its journey at the *same* depot;

(iii) some side constraints on the number of routes and on their configurations are satisfied.

Before describing these constraints, it seems appropriate to bring some precision to the first condition. This condition is irrelevant when C satisfies the triangle inequality: in this case, it is never advantageous for a vehicle to deviate from the shortest route between any two cities and thus to visit any city more than once. However, most road networks are incomplete and multiple passages through the same city are more often than not necessary. This difficulty can be circumvented as follows. First replace each c_{ij} by the length of the shortest path from i to j and then establish the following distinction: each city must be *serviced* exactly once but may be *crossed* as many times as necessary if it lies on many shortest paths. However, it is sometimes inefficient to proceed this way: in some algorithms (see for example [27,28,65]), it is preferable to take advantage of the sparsity of the graph and to use the original c_{ij}'s while requiring that each city be visited *at least* once. Some examples of this approach will be presented.

The most common side conditions include:

(i) Capacity restrictions: a non-negative weight d_i is attached to each city i of N-R. (For $i \in R$, assume that $d_i = 0$.) The sum of weights of any vehicle route may not exceed that vehicle capacity (see for example [7,10,27,28,33,42,49,55]). (Capacitated VRPs will be referred to as CVRPs.)

(ii) The number of cities on a route may be bounded above (this constitutes a particular case of (i)).

(iii) Total time restrictions: the length of the route of any vehicle may not exceed a prescribed bound L; this length is made up of intercity travel times and of stopping times (δ_i) at each city i included on the route [50,55]. (Time or distance constrained VRPs will be referred to as DVRPs.)

(iv) Time windows: city i may have to be visited within the time interval $[a_i,b_i]$ (this includes the two cases $a_i = -\infty$ and $b_i = \infty$) [17-20,69,71,72].

(v) Precedence relations between pair of cities: city i may have to be visited before city j [73].

This list is by no means exhaustive and it is beyond the scope and purpose of this paper to describe every possible ramification of the VRP. This survey concentrates on formulations and methods for the more basic models, mentioning, where appropriate, important extensions for which exact algorithms have been developed. The following restrictions will be made.

(i) Fixed costs on depots and vehicles are not generally considered: their eventual inclusion in the models is in most cases straightforward.

(ii) Unless otherwise specified (see section 4), there is only one depot located at city 1.

(iii) In most cases, all vehicles are identical and have the same capacity D.

(iv) All vehicles are assumed to drive at the same constant speed; in such a context, the c_{ij}'s may be interpreted as travel times whenever appropriate. The c_{ij}'s sometimes represent travel costs.

1.2 Classification of methods

It appears that all known exact algorithms for the VRP can be classified into one of the following categories:

(i) direct tree search methods,

(ii) dynamic programming (DP),

(iii) integer linear programming (ILP).

The latter category is very broad and accounts for most of the research effort of recent years. It will be subdivided into three sections, according to the classification suggested by Magnanti [60]:

iiia) Set partitioning formulations.

iiib) Vehicle flow formulations (by far the most widely used).

iiic) Commodity flow formulations.

Several interesting relationships between these three LP formulations have been outlined by Magnanti. Readers are referred to [60] for full details.

2. DIRECT SEARCH ALGORITHMS

2.1 Branching on arcs

Direct search methods consist of sequentially building vehicle routes by means of a branch and bound tree. The paper by Christofides and Eilon [7] published in 1969 contains one of the earliest descriptions of such a method. It is based on the Little et al. algorithm [59] for the TSP and applies to CVRPs and DVRPs.

It will be useful to first present an ILP formulation of the TSP. Let x_{ij} $(i \neq j)$ be a 0-1 variable indicating whether arc (i,j) appears $(x_{ij} = 1)$ or not $(x_{ij} = 0)$ in the optimal solution and let $\bar{S} = N-S$. Then the VRP can be formulated as follows:

(TSP1) minimize $\sum_{i,j \in N} c_{ij} x_{ij}$

subject to

(2) $\sum_{\substack{i \in N \\ i \neq j}} x_{ij} = 1 \qquad (j \in N)$

(3) $\sum_{\substack{j \in N \\ i \neq j}} x_{ij} = 1 \qquad (i \in N)$

(4) $\sum_{i \in S, j \in \bar{S}} x_{ij} \geq 1 \qquad (2 \leq |S| \leq n-2,\ S \subset N)$

(5) $x_{ij} = 0, 1 \qquad (i \neq j,\ i,j \in N)$

This formulation was first proposed by Dantzig et al. [14]. Constraints (2), (3) and (5) describe an assignment problem while constraints (4) ensure that the graph associated with the optimal solution

is connected. When constraints (2) and (3) are satisfied, constraints (4) are equivalent to

$$(6) \qquad \sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad (2 \leq |S| \leq n-2,\ S \subset N)$$

This equivalence can be extended to more general situations (see [68]).

Two well known relaxations of the TSP are

(i) the assignment problem (AP) characterized by constraints (2), (3), (5);

(ii) the shortest spanning 1-tree problem satisfying constraints (4) and (5). Efficient algorithms exist for solving the shortest spanning 1-tree [22,45].

Thus, a valid bound on the TSP can be derived by solving either of these two problems or by computing a lower bound on their solutions. Comparisons of these two bounds are given in [24].

In order to apply a TSP algorithm to the VRP, it is convenient to first transform the distance matrix C. This can be done by adjoining $\bar{m}$-1 artificial depots to the set N of cities, where $\bar{m}$ is an upper bound on m satisfying

$$(7) \qquad \bar{m} \geq \left\lceil \frac{\sum_{i \in N} d_i}{D} \right\rceil$$

Here and elsewhere, $\lceil t \rceil$ denotes the smallest integer greater than or equal to t if $t > 0$; (it is equal to 1 if $t \leq 0$).

Let N* be the set of depots:

(8) $N^* = \{1, n+1, \ldots, n+\bar{m}-1\}$

and let $N' = N \cup N^*$.

The extended cost matrix $C' = (c'_{ij})$ is then defined by

$$(9) \quad c'_{ij} = \begin{cases} c_{ij} & (i,j \in N) \\ c_{i1} & (i \in N,\ j \in N^*) \\ c_{1j} & (i \in N^*,\ j \in N) \\ \lambda & (i,j \in N'-N) \end{cases}$$

In [7], λ is set equal to infinity in order to prohibit inter-depot arcs from entering the solution. Lenstra and Rinnooy Kan [57] have later observed that $-\lambda$ can be interpreted as the cost of using a vehicle. Therefore,

$\lambda=\infty$ yields the minimum distance for $\bar{m}$ vehicles;

$\lambda=0$ yields the minimum distance for at most $\bar{m}$ vehicles;

$\lambda=-\infty$ yields the minimum distance for the minimum number of vehicles.

Any *sequential* algorithm for the TSP, i.e. an algorithm which successively fixes arcs (variables) at 0 or 1 in a branch and bound tree (cf. Little et al. [59]) can be used for the VRP with the following modifications. Christofides and Eilon [7] fathom a branch in one of these cases:

(i) The total load for a vehicle exceeds its capacity D.

(ii) The total distance accumulated for a vehicle exceeds L.

(iii) The total weight of cities not already included in the tour exceeds the total capacity of the yet unused vehicles.

Instead of computing the assignment bound at every node of the search tree (as in [59]), Christofides and Eilon suggest using the shortest spanning 1-tree bound.

The algorithm was tested on two VRPs involving capacity restrictions only: a 6-city problem taken from [42] and the Dantzig and Ramser 13-city example [15], were solved in 1.5 and 5 minutes respectively on an IBM 7090.

2.2 Branching on routes

In the previous method, branches of the tree were created by including an arc in the solution or by excluding it from consideration. In [6], Christofides describes a depth first branch and bound algorithm based on a different philosophy. The class of VRPs to which the algorithm applies is very wide: problems can include capacity or distance restrictions, time windows, precedence conditions, stopping times, etc. In fact, the method perform better on tight problems as less branches need be explored.

Each level of the search tree corresponds to the definition of a new vehicle route. Thus the tree will have at most $\bar{m}$ levels. To each node of the tree, corresponds a set of vehicle routes. Tree construction can be summarized as follows. Let g represent a node of the search tree and define

$F(g)$: the set of unrouted (free) cities beyond node g;

$\bar{F}(g) = N-F(g)$: the set of routed cities at node g.

Forward branching from g is done by considering a city i of F(g) and by generating the list of all feasible routes including i. Each of these routes corresponds to an immediate descendant node g'. It is apparent that it is advantageous to choose i in such a way that as few routes as possible are generated. The following criteria are used to eliminate from consideration potential routes passing through i:

(i) The route containing i violates at least one side constraint.

(ii) The unused vehicles cannot feasibly supply cities in F(g') (the total weight of these cities exceeds the total unused vehicle capacity).

(iii) Define z^* : the cost of the best known feasible solution;
$c(S)$: the optimal cost of supplying cities of S ($S \subseteq N$);
$\underline{c}(S)$: a lower bound on $c(S)$;
$\bar{c}(S)$: an upper bound on $c(S)$.

Then node g' need not be created if

$$(10)\quad c(\bar{F}(g)) + c(\bar{F}(g') - \bar{F}(g)) + \underline{c}(F(g')) \geq z^*.$$

(iv) g' is dominated by another immediate descendant g" to node g, i.e.

$$(11)\quad c(\bar{F}(g") - \bar{F}(g)) + \bar{c}(F(g")) \leq c(\bar{F}(g') - \bar{F}(g)) + \underline{c}(F(g'))$$

Note that c can be replaced by $\underline{c}$ in the LHS of (10) and in the RHS of (11), and by $\bar{c}$ in the LHS of (11).

If the VRP involves capacity or total distance constraints only, $c(S)$ may be computed by solving a TSP on S. Similarly, $\bar{c}(S)$ can be determined by using or adapting a TSP heuristic [37, 39] and the value of $\underline{c}(S)$ can be obtained from one of the known TSP bounds [24]. However, it is harder to determine these values in problems with time windows.

The largest problem reported solved to optimality with the algorithm contains 31 cities. No details are provided on the type of constraints or computing times.

2.3 k-degree centre trees and q-routes

The efficiency of any branch and bound algorithm (for a minimization problem) rests on the computation, at every node of the search tree, of sharp lower bounds on the value of the optimum. In [9], Christofides, Mingozzi and Toth derived such bounds for the VRP. The problem they consider is a VRP with exactly m vehicles and with capacity and time restrictions. In addition to travel times c_{ij} between cities, the authors also consider an unloading time δ_i at each city i. The problem is formulated as a "three-index vehicle flow program" (see section 5) but no direct use is made of this model either in the computation of the bounds or in the algorithm which is a direct search procedure.

The first of these bounds is based on the fact that the value of the optimal VRP solution is bounded below by the total length of a *k-degree centre tree* (k-DCT), i.e. a tree where the degree of node 1 is k. Consider E, the set of all arcs. At the optimum, the set of arcs belonging to the solution can be partitioned into 3 sets:

(i) E_1 : arcs forming a k-DCT (where $k = 2m-y$ and $y \leq m$);

(ii) E_2 : y arcs incident to node 1;

(iii) E_3 : m-y arcs not incident to node 1.

Let ξ_ℓ^t ($t = 1, 2, 3$; $\ell \in E$) be 0-1 variables taking the value 1 if and only if arc ℓ belongs to E_t in the optimal solution, let c_ℓ be the cost of arc ℓ, E^i the set of all arcs incident to node i and $(S,\bar{S})$, the set of all arcs with one vertex in S and the other vertex in its complement $\bar{S}$. Then the VRP is

$$\text{(VRP1)} \quad \text{minimize} \quad \sum_{\ell \in E} c_\ell \, (\xi_\ell^1 + \xi_\ell^2 + \xi_\ell^3)$$

subject to

$$(12)\quad \sum_{\ell \in (S,\bar{S})} \xi_\ell^1 \geq 1 \qquad (S \subset N,\ S \neq \phi)$$

$$(13)\quad \sum_{\ell \in E^1} \xi_\ell^1 = 2m - y$$

$$(14)\quad \sum_{\ell \in E} \xi_\ell^1 = n - 1$$

$$(15)\quad \sum_{\ell \in E_1} \xi_\ell^2 = y$$

$$(16)\quad \sum_{\ell \in E-E^1} \xi_\ell^3 = m - y$$

$$(17)\quad \sum_{\ell \in E^i} (\xi_\ell^1 + \xi_\ell^2 + \xi_\ell^3) = 2 \qquad (i = 2, \ldots, n)$$

$$(18)\quad \xi_\ell^t = 0, 1 \qquad (t = 1, 2, 3;\ \ell \in E)$$

$$(19)\quad y \geq 0 \text{ and integer}$$

Three problems are extracted from (VRP1) by only considering variables ξ_ℓ^t (t=1, 2, 3). In each case, constraints (17) are incorporated into the objective function in a Lagrangean fashion to yield

$$(20)\quad V^t(\lambda, y) = \sum_{\ell \in E} (c_\ell + \lambda_{\alpha(\ell)} + \lambda_{\beta(\ell)}) - 2 \sum_{i=1}^{n} \lambda_i$$

where $\alpha(\ell)$, $\beta(\ell)$ are the two terminal vertices of arc ℓ, $\lambda = (\lambda_i)$ is the vector of non-negative penalties associated with constraints (17) and $\lambda_1 = 0$. A lower bound for the solution of (VRP1) is therefore

$$(21)\quad LB1 = \max_{m_1 \le y \le m} \{\max_{\lambda \ge 0} \sum_{t=1}^{3} V^t(\lambda, y)\}$$

where m_1 represents a lower bound on the number of single customer tours in the optimal solution. Taking into account capacity and maximum time constraints, m_1 can be chosen so as to satisfy the following conditions:

(i) Suppose the cities are ordered in decreasing order of the d_i's. Then we must have

$$(22)\quad (m - m_1)\, D \ge \sum_{i=m_1+1}^{n} d_i \quad .$$

(ii) Similarly, every city i contributes an amount of at least $u_i = \delta_i + \frac{1}{2}(c_{ii_1} + c_{ii_2})$ to the length of a route where i_1 and i_2 are the two cities nearest to i. Then, if the cities are ordered in decreasing order of the u_i's, m_1 must satisfy

$$(23)\quad (m - m_1)\, L \ge \sum_{i=m_1+1}^{n} u_i .$$

Christofides et al. describe an efficient procedure to compute (21).

The second bound developed in [9] is based on *q-routes*, used by Houck et al. [44] in the case of the TSP. The bound, which applies to CVRPs, is derived as follows. Let W be the set of all possible weights that could exist on any vehicle route. Let the elements of W be ordered in ascending order and let $q(\ell)$ be the value of the ℓ^{th} element of W. Define $\psi_\ell(i)$ as the value of the least cost route

(i) passing through i;

(ii) starting and ending at the depot;

(iii) having no loop of the form (i_1, i_2, i_1);

(iv) having a total weight $q(\ell)$.

Such a route is called a q-route. Then, Christofides et al. prove that LB2 is a valid lower bound for the CVRP:

$$(24)\quad LB2 = \sum_{i=1}^{n} \min_{\ell=1,\ldots,|W|} \left(\frac{\psi_\ell(i)\, d_i}{q(\ell)}\right)$$

The authors show that $\psi_\ell(i)$ can be computed through a relatively simple recursion procedure. It is obvious that the effort involved in the computation of LB2 is directly related to $|W|$.

These two bounds (based on k-DCTs and on q-routes) were embedded in branch and bound algorithms and applied to CVRPs ranging from 10 to 25 cities. In the case of the first bound, two branching strategies were used: branching on arcs and branching on routes; with the second bound, only the first strategy was used. The results indicate that LB2 is generally superior to LB1: it certainly produces much sharper bounds at the root of the search tree. However, the best algorithm overall used LB1 in conjunction with a "branching on routes" strategy. All test problems were solved within 250 seconds on a CDC 7600 computer, using an FTN compiler.

3. DYNAMIC PROGRAMMING

Dynamic programming (DP) has been applied to several types of VRPs. First, consider the general formulation provided by Eilon et al. [24].

3.1 General dynamic programming formulation for the VRP

Consider a VRP with a fixed number $\bar{m}$ of vehicles. Let c(S) denote the cost of the optimal single route through the depot (city 1) and all the customers of a subset S of N-{1}. We wish to minimize

$$(25) \quad z = \sum_{j=1}^{\bar{m}} c(S_j)$$

over all feasible partitions $\{S_1, \ldots, S_{\bar{m}}\}$ of N-{1}. Let $f_k(U)$ be the minimum cost achievable using k vehicles and delivering to a subset U of N-{1}. Then the minimum can be determined through the following recursion formula:

$$(26) \quad f_k(U) = \begin{cases} c(U) & (k = 1) \\ \min\limits_{U^* \subseteq U \subseteq N-\{1\}} [f_{k-1}(U-U^*) + c(U^*)] & (k > 1) \end{cases}$$

The solution cost is equal to $f_{\bar{m}}(N-\{1\})$ and the optimal partition will correspond to the optimizing subsets U* in (26).

It is apparent that if $f_k(U)$ has to be computed for all values of k and for all subsets U of N-{1}, the number of computations required by the method is likely to be formidable in all but very small size problems. Efficient methods based on dynamic programming require a substan-

tial reduction of the number of states by means of feasibility or dominance criteria. Consider for example the CVRP. In this problem, U and U* must satisfy

$$(27)\quad \sum_{i\in N-\{1\}} d_i - (\bar{m} - k)\, D \le \sum_{i\in U} d_i \le k\, D \qquad (k=1, \ldots, \bar{m})$$

and

$$(28)\quad \sum_{i\in U} d_i - (k - 1)\, D \le \sum_{i\in U^*} d_i \le D \qquad (k=1, \ldots, \bar{m})$$

3.2 State-space relaxation

State-space relaxation provides another efficient means of reducing the number of states. The method was introduced by Christofides et al. [10]. It provides a lower bound $\underline{z}^*$ on the cost z^* of the optimal solution. The optimum can then be reached by embedding the method in a branch and bound procedure.

Consider the general DP recursion

$$(29)\quad f_{0,i}(0,j) = \min_{k\in\Delta^{-1}(j)} [f_{0,i-1}(0,k) + c_i(k,j)]$$

where

$f_{0,i}(0,j)$ is the least cost of changing the system from state 0 at stage 0 to state j at stage i

$\Delta^{-1}(j)$ is the set of all possible states from which state j can be reached directly

$c_i(k,j)$ is the cost of changing the system from state k at stage i-1 to state j at stage i

Let $g(\cdot)$ be a mapping from the state space S associated with (29) to a state space G with smaller cardinality and let $F^{-1}(g(j))$ be a set satisfying

$$(30)\quad k \in \Delta^{-1}(j) \Rightarrow g(k) \in F^{-1}(g(j))$$

Recursion (29) then becomes

$$(31)\quad f_{0,i}(g(0),g(j)) = \min_{t \in F^{-1}(g(j))} [f_{0,i-1}(g(0),t) + \bar{c}_i(t,g(j))]$$

where

$$(32)\quad \bar{c}_i(t,g(j)) = \min [c_i(k,\ell): g(k) = t,\ g(j) = g(\ell)]$$

It results that

$$(33)\quad f_{0,i}(g(0),g(i)) \leq f_{0,i}(0,i)$$

This relaxation is useful only if

(i) $F^{-1}(\cdot)$ can be easily determined; this will be so if $g(\cdot)$ is *separable* so that given $g(U)$ and r, $g(U-\{r\})$ can be computed;

(ii) $g(\cdot)$ must be such that the optimization of (32) is over a small domain or that a good lower bound on $\bar{c}_i(t,g(j))$ can be computed.

Using various formulations and several choices of $g(\cdot)$, Christofides et al. have applied state-space relaxation to the TSP and to a variety of VRPs. Some of the results they have obtained are reported in sections 3.3 a) and b).

3.3 Application of dynamic programming to some VRPs

DP has been used successfully to solve some VRPs to optimality or to obtain very sharp bounds on the value of their optimal solutions. Here are some examples.

a) *Capacitated VRP (CVRP)* - Christofides et al. [10] present three formulations for this problem. The first one is provided by (26), subject to (27) and (28). Of the other two, one has been used to obtain a good lower bound on the optimum. It can be described as follows. Let $f_k(U,r)$ be the least cost of supplying a set U of customers, using only k vehicles, with the last customers of the corresponding k routes being among the customers 2, ..., r ($k \leq r \leq n$). Let $c(U,r)$ be the cost of the TSP solution through $U \cup \{1\}$ where the last customer (before the depot) is r. The recursion is then

$$(34)\quad f_k(U,r) = \begin{cases} \min \left[f_k(U,r-1),\ \min_{U^* \subset U} \{ f_{k-1}(U-U^*,\ r-1) + c(U^*,r) \} \right] & (k,r > 1) \\ c(U,r) & (k = 1) \end{cases}$$

subject to (27).

For this problem, the mapping function is defined by

$$(35)\quad g(U) = \sum_{i \in U} d_i$$

Recursion (21) then becomes

$$(36)\quad f_k(g(U),r) = \min \left[f_k(g(U),r-1),\ \min_{p} \{ f_{k-1}(g(U)-p,r-1) + \bar{c}(p,r) \} \right]$$

subject to

$$(37)\quad g(N) - (\bar{m}-1)\, D \le p \le \min\,(g(U), D)$$

Using this relaxation, lower bounds on the optimal VRP solutions were obtained for 10 problems containing from 10 to 25 cities. For the 10-city problem, the bound was in fact equal to the optimum. Otherwise, the ratio "lower bound ÷ optimum" varied between 93.1% and 99.6%.

b) *TSP with time windows* (Christofides et al. [10]) Consider a TSP in which r_i time windows $[a_i^k, b_i^k]$ $(k=1, \ldots, r_i)$ are associated with each city i. Without loss of generality, it is assumed that the time windows are disjoint and ordered so that

$$(38)\quad b_i^{k-1} < a_i^k \qquad (k=2, \ldots, r_i)$$

$$b_i^0 = 0$$

Also assume that there is a *processing time* δ_i at city i and that c_{ij} represents a *travel time* from i to j. Let t_i be the time of arrival at city i in the optimal solution. Then we seek the shortest Hamiltonian circuit through the n cities under the constraint

$$(39)\quad a_i^k \le t_i \le b_i^k \qquad \text{for some } k \in \{1, \ldots, r_i\}$$

In this problem, the efficiency of the DP approach is largely dependent on the number of states which can be eliminated through feasibility considerations: tighter time windows will lead to the generation of fewer states. Here again the authors use state-space relaxation; the particular mapping $g(\cdot)$ which is used is not mentioned. The

authors report result for 40 "moderately tight" problems ranging from 15 to 50 cities. The method only provides a lower bound on the optimum. For one particular 30 city problem, this bound attains 99.8% of the optimum.

c) *Dial-a-ride problems* - This class of problems can be described as follows. Consider n groups of customers requesting to be picked up at a given location and to be delivered to another location. In *many-to-many* problems, origins as well as destinations are all distinct points. In *many-to-one* problems, all delivery points coincide. What follows is valid for many-to-many problems. Let {1, ..., n} be the set of origins and {n+1, ..., 2n}, the set of destinations. There is a depot located at node 0. Travel times t_{ij} and distances c_{ij} between pairs of cities are given. In addition, there is a processing time δ_i at each node i. The problem consists of establishing optimal vehicle routes subject to some operating rules and constraints.

There are two main classes of operating rules. In the *static case*, customers request service and these requests are registered in a list. At a certain point T_0 in time, a vehicle becomes available and the task is to provide service to all customers in the list at that time; no new customers may be added to a route during its execution. In the *dynamic case*, new customers become eligible for immediate inclusion in a route as they make their request.

The most common constraints are

(i) Vehicle capacities: at node i are associated d_i customers to be picked up (if $d_i > 0$) or delivered (if $d_i < 0$).

Assume that

$$(40)\quad d_i > 0 \qquad (i=1, \ldots, n)$$

$$d_i = -d_{i-n} \qquad (i=n+1, \ldots, 2n)$$

At most $\bar{m}$ vehicles of capacity D are used for the operation. At no time can vehicle capacity be exceeded.

(ii) Time windows: with each node i is associated a unique time window $[a_i, b_i]$. Vehicles may not leave i after b_i; they are however allowed to stay idle at the node if they arrive before a_i.

(iii) Maximum position shifts: these were described by Psaraftis [71] in the following terms.

> "In any particular vehicle route we can identify the sequence of pickups and the sequence of deliveries, sequences which, in general, will be *merged* with one another. The position (1st, 2nd, etc.) that a particular customer holds in the sequence of pickups will not in general be the same as his First-Come-First-Served (FCFS) position in the initial list of customer requests, the difference of these two positions constituting the *pickup position shift* of that customer. For instance, a customer holding the 5th position in the initial list and being picked up 3rd has a pickup position shift of +2 while this shift becomes -1 if the customer is picked up 6th. A *delivery position shift* can be similarly defined as the difference between the position of a customer in the sequence of deliveries and the FCFS position of that customer in the initial list of requests."

Then, for any customer, neither of these two position shifts may exceed a prescribed *maximum position shift*.

Several variants of the problem can then be defined. In [71], Psaraftis considers the static and the dynamic cases with capacity and maximum position shift constraints. The objective function is a linear combination of

(i) route durations;

(ii) customers waiting times (between T_0 and pick up times);

(iii) customers riding times.

He proposes an exact $O(n^2 3^n)$ exact algorithm, an extension of the classical Held and Karp [43] DP algorithm for the TSP. Taking advantage of the fact that many infeasible states need not be defined, Psaraftis succeeds in solving optimally problems involving up to 9 customers in the static case. No computational results are reported in the dynamic case.

Psaraftis later described a similar algorithm [72] for problems in which maximum position shift constraints are replaced by time windows. No computational results were reported, but the author stated that he did not expect the CPU times to be significantly different from those obtained in [71].

Desrosiers et al. [18] consider the static problem where the objective is to minimize the total distance travelled while respecting vehicle capacities and time windows. The method is based upon an earlier algorithm by the same authors for the shortest path problem with time windows [16,19,20]. In this method, there are two labels associated with each state, one representing accumulated travelled distance, the other, elapsed time. As in the above methods, states which cannot feasibly be part of the solution are not defined. Furthermore, states can be eliminated through dominance criteria. For example, let r_i be the distance label at state i and t_i be the time label at the same state. Consider two states i and j such that $r_i \leq r_j$ and $t_i \leq t_j$. Then state j can be eliminated.

Using DP and dominance criteria, Desrosiers et al. have solved optimally several problems involving up to 40 requests in less then 6 seconds on the University of Montreal CYBER 173.

In a different paper [17], the same authors consider the multi-vehicle dial-a-ride problem. They use a decomposition approach in which the subproblems are single vehicle problems. Each of these subproblems is solved optimally although the global solution is not optimal. Real-life problems involving up to 880 requests were solved.

4. SET PARTITIONING FORMULATIONS

Over the years, several integer linear programming formulations have been suggested for the VRP. Among these, set partitioning formulations cover a wide range of problems. Unfortunately, due to the large number of variables they contain, it will rarely be practicable to use them to derive optimal solutions. One interesting exception is the algorithm developed by Desrosiers et al. [21] for school bus routing.

Balinski and Quandt [1] were among the first to suggest such a formulation. Consider all feasible routes j and let a_{ij} be a 0-1 coefficient taking the value 1 if and only if city i appears on route j. Let c_j^* be the optimal cost of route j and x_j be a 0-1 variable equal to 1 if and only if route j is used in the optimal solution. Then the problem is to

(VRP2) minimize $\sum_j c_j^* x_j$

subject to

(41) $\sum_j a_{ij} x_j = 1$ $\qquad (i \in N-\{1\})$

(42) $x_j = 0, 1$ $\qquad$ (for all j)

This formulation can easily be transformed to include the case where multiple passages through city i are allowed: it suffices to replace the equality sign of constraint i by a "$\geq$" sign. Similarly, the number of passages through city i may be set equal to a number greater than 1.

These are two major difficulties associated with this approach.

(i) The large number of binary variables which can run into the millions in most real-life cases. Only in extremely constrained pro-

blems (i.e. in problem very few feasible solutions) will the number of variables be small enough to enable the problem to be solved directly.

(ii) The difficulty in computing the c_j^* values. For example, in the CVRP, each route j corresponds to a set of cities S_j satisfying

$$(43) \quad \sum_{i \in S_j} d_i \leq D$$

The value of c_j^* is then obtained by solving a TSP on S_j. As such, this formulation offers no obvious advantage over the general DP formulation.

However, if the number of variables is relatively small and if the objective is to minimize the number of vehicles, i.e. $c_j^* = 1$ for all j, the linear relaxation of (VRP2) often provides an integer solution [76]. If the solution (x^*) is fractional and gives a non-integral objective function, then the following cutting plane can be introduced

$$(44) \quad \sum_j x_j \geq \lceil \sum_j x_j^* \rceil$$

Very few cuts are generally required to reach integrality [69].

Various authors have proposed column generation schemes to solve VRPs by the set partitioning approach. Rao and Zionts [74] consider the problem of establishing m vessel routes between ports. At each iteration, m shortest complete cycles (one for each vessel) are generated by means of the out-of-kilter algorithm [29] and the corresponding set partitioning problem is solved. New columns are introduced into the problem to replace old ones as long as they can reduce the value of the objective function. Unfortunately, no computational results are reported for this method.

Foster and Ryan [30] also suggest a column generation approach in which the routes are obtained by dynamic programming. These authors report that the method, although not run to optimality, provides good routes when compared to those obtained by previous researchers [7,32,36, 80].

Orloff [69] discussed the use of column generation in the context of school bus scheduling and suggested some solution approaches. Desrosiers et al. [21] developed an efficient column generation algorithm for the same problem. In order to summarize their work, some definitions and notation will be required.

Consider a set *trips* to be covered by buses and a set of *intertrips* corresponding to unproductive bus journeys between the trips. Every trip i is characterized by a place of origin, a destination, a duration, a cost and a time interval $[a_i,b_i]$ during which the trip must begin. The intertrip arc (i,j) goes from the end of trip i to the beginning of trip j. Its duration t_{ij} and its cost c_{ij} include respectively the duration and cost of trip i. A *route* is a sequence of trips and intertrips carried out by the same vehicle. The problem is to determine routes and schedules for all the trips so as to minimize the number of vehicles and travel costs for that number of vehicles, while respecting network and scheduling constraints.

The problem is formulated as (VRP2) where i represents a trip and j, a route. A column generation scheme is used and columns are provided by the solution of the following shortest path problem with time windows. First define the following notation.

P is the set of trips

I is the set of intertrips arcs

s is a source and t is a sink

$P' = P \cup \{s,t\}$

$A = P \cup (\{s\} \times P) \cup (P \times \{t\})$

$x_{ij} = 1$ if and only if (i,j) is used by a vehicle

t_i is the starting time of trip i

u_i is the i^{th} dual variable of the linear relaxation of (VRP2).

Then the problem is:

$$\text{(SPTW)} \quad \text{minimize} \quad \sum_{(i,j)\in A} (c_{ij} - u_i)\, x_{ij}$$

subject to

$$\text{(45)} \quad \sum_{j\in P'} x_{ij} = \sum_{j\in P'} x_{ji} \qquad (i \in P)$$

$$\text{(46)} \quad \sum_{j\in P} x_{sj} = \sum_{j\in P} x_{jt} = 1$$

$$\text{(47)} \quad x_{ij} > 0 \Rightarrow t_i + t_{ij} \le t_j \qquad (i,j) \in I$$

$$\text{(48)} \quad a_i \le t_i \le b_i \qquad (i \in P)$$

$$\text{(49)} \quad x_{ij} = 0, 1 \qquad ((i,j) \in A)$$

In this formulation, constraints (45) are flow conservations equations and constraint (46) expresses the fact that one vehicle leaves and enters the depot. Constraints (47) ensure that trip j can only be made immediately after trip i if this succession does not violate a time window constraint. It can be shown [21] that (SPTW) possesses an integer solution even if constraints (49) are relaxed (i.e.

the bounds on the variables are relaxed as well as the integrality conditions). Desrosiers et al. solve (SPTW) by means of a dynamic programming algorithm [19]. The authors suggest introducing several routes (columns) simultaneously into the master problem (VRP2), as opposed to only one at a time. The effect of this strategy is to reduce overall computation times by a factor of 2. The master problem is also solved by relaxing the integrality requirements and by introducing two cuts in order to eliminate solutions with a fractional cost or a fractional number of vehicles.

Desrosiers et al. report exact solutions for problems involving up to 151 trips. The running time for the largest problems varies between 94 seconds and 376 seconds on the University of Montreal CYBER 173 computer, in the case where time windows are relatively narrow. As expected, computation times increase as time windows widen.

5. VEHICLE FLOW FORMULATIONS

Most ILP based algorithms for vehicle routing problems are based on vehicle flow formulations. These formulations use binary variables to indicate whether a vehicle travels between two given cities in the optimal solution. We distinguish two families of vehicle flow formulations: three-index formulations and two-index formulations. In the first case, three indices are attached to each flow variable: the origin, the destination and the vehicle making the trip; in the second case, the vehicle is not identified.

5.1 Three-index formulations

The following formulation is adapted from the work of Golden et al. [38] for asymmetrical VRPs involving at most $\bar{m}$ vehicles and r depots. In addition to the standard notation, define

D_k : the capacity of vehicle k

t_{ij}^k : the time taken by vehicle k to travel from i to j ($t_{ii} = \infty$)

δ_i^k : the time required for vehicle k to service city i ($\delta_i^k = 0$, $i \in R$)

L_k : the maximum allowed length of route k

c_{ij}^k : the cost of using vehicle k from i to j

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle k travels directly from i to j} \\ 0 & \text{otherwise} \end{cases}$$

The VRP can be formulated as follows.

$$\text{(VRP3)} \quad \text{minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{\bar{m}} c_{ij}^k \, x_{ij}^k$$

subject to

$$(50)\quad \sum_{i=1}^{n} \sum_{k=1}^{\bar{m}} x_{ij}^{k} = 1 \qquad (j = r+1, \ldots, n)$$

$$(51)\quad \sum_{j=1}^{n} \sum_{k=1}^{\bar{m}} x_{ij}^{k} = 1 \qquad (i = r+1, \ldots, n)$$

$$(52)\quad \sum_{i=1}^{n} x_{i\ell}^{k} - \sum_{j=1}^{n} x_{\ell j}^{k} = 0 \qquad (k=1, \ldots, \bar{m};\ \ell=1, \ldots, n)$$

$$(53)\quad \sum_{i=1}^{n} d_i \left(\sum_{j=1}^{n} x_{ij}^{k} \right) \leq D_k \qquad (k=1, \ldots, \bar{m})$$

$$(54)\quad \sum_{i=1}^{n} \delta_i^k \sum_{j=1}^{n} x_{ij}^{k} + \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij}^{k} x_{ij}^{k} \leq L_k \qquad (k=1, \ldots, \bar{m})$$

$$(55)\quad \sum_{i=1}^{r} \sum_{j=r+1}^{n} x_{ij}^{k} \leq 1 \qquad (k=1, \ldots, \bar{m})$$

$$(56)\quad \sum_{j=1}^{r} \sum_{i=r+1}^{n} x_{ij}^{k} \leq 1 \qquad (k=1, \ldots, \bar{m})$$

(57) The solution may not contain subtours which do not include a depot.

$$(58)\quad x_{ij}^{k} = 0, 1 \qquad (i,j=1, \ldots, n;\ k=1, \ldots, \bar{m})$$

In this formulation, constraints (50) and (51) specify that each city must be served exactly once by one and only one vehicle; constraints (52) ensure that every city is entered and left by the same vehicle; constraints (53) guarantee that vehicle capacities are never exceeded while constraints (54) are imposed in order to ensure that no vehicle route exceeds its time limit. Constraints (55) and (56) ensure that no more than $\bar{m}$ vehicles leave the depots and there is at most one vehicle per depot.

Subtour elimination constraints specified by (57) may be one of the following:

$$(57a)\quad \sum_{i\in S}\sum_{j\notin S} x_{ij}^k \geq 1 \qquad (|S| \geq 1;\ R \subseteq S \subset N;\ k=1, \ldots, \bar{m})$$

$$(57b)\quad \sum_{i\in S}\sum_{j\in S} x_{ij}^k \leq |S| - 1 \qquad (|S| \geq 1;\ S \subseteq N - R;\ k=1, \ldots, \bar{m})$$

$$(57c)\quad y_i - y_j + n\, x_{ij}^k \leq n - 1 \qquad (k=1, \ldots, \bar{m};\ r+1 \leq i \neq j \leq n;\ \text{for some } y_i, y_j \in \mathbb{R})$$

Constraints (57a) and (57b) were first suggested by Dantzig et al. [14] for the TSP while constraints (57c) are those of Miller, Tucker and Zemlin [66].

It is relatively easy to include fixed vehicle costs f_k into the objective function by adding the term $\sum_{k=1}^{\bar{m}} f_k \left(\sum_{i=1}^{r} \sum_{j=r+1}^{n} x_{ij}^k \right)$. It is also possible to impose the restriction that city i be visited within the time interval $[a_i, b_i]$ $(i=r+1, \ldots, n)$. For this, define a variable t_i equal to the arrival time at city i and impose

$$(59)\quad t_i = 0 \qquad (i=1, \ldots, r)$$

$$(60)\quad a_i \leq t_i \leq b_i \qquad (i=r+1, \ldots, n)$$

$$(61)\quad \begin{cases} t_j \geq (t_i + \delta_i^k + t_{ij}^k) - (1 - x_{ij}^k)\, T \\ t_j \leq (t_i + \delta_i^k + t_{ij}^k) + (1 - x_{ij}^k)\, T \end{cases} \qquad (i,j=1, \ldots, n;\ k=1, \ldots, \bar{m})$$

Where T is an arbitrarily large number.

Fisher and Jaikumar [25] present a different three-index vehicle flow formulation for the single depot VRP with capacity restrictions, time windows and a fixed number $\bar{m}$ of vehicles. The authors have then developed an efficient algorithm based on this formulation. Although the algorithm seems to have been used only to provide a heuristic solution to the problem, as do some other approximate methods, it guarantees an optimal solution in a finite number of steps, if run to completion. The approach developed by Fisher and Jaikumar also possesses many other desirable features.

The notation is partly borrowed from (VRP3). Extra variables are introduced:

$$y_{ik} = \begin{cases} 1 & \text{if city } i \text{ is served by vehicle } k \\ 0 & \text{otherwise} \end{cases}$$

but travel costs and times are vehicle independent (although these could easily be taken into account by the model and algorithm). There are no service times at the nodes. The formulation is

(VRP4) minimize $\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{\bar{m}} c_{ij} x_{ij}^{k}$

subject to

(62) $\sum_{i=1}^{n} d_i y_{ik} \leq D_k$ $\qquad (k=1, \ldots, \bar{m})$

(63) $\sum_{k=1}^{\bar{m}} y_{ik} = \begin{cases} \bar{m} & (i = 1) \\ 1 & (i=2, \ldots, n) \end{cases}$

(64) $\sum_{i=1}^{n} x_{ij}^{k} = y_{jk}$ $\qquad (j=1, \ldots, n;\ k=1, \ldots, \bar{m})$

(65) $\sum_{j=1}^{n} x_{ij}^{k} = y_{ik}$ $\qquad (i=1, \ldots, n;\ k=1, \ldots, \bar{m})$

$$(66)\quad \sum_{i,j\in S} x_{ij}^{k} \le |S| - 1 \qquad (S \subset N;\ 2 \le |S| \le n-1;\ k=1, \ldots, \bar{m})$$

$$(67)\quad \begin{cases} t_j \ge t_i + t_{ij} - (1 - x_{ij}^{k})\, T \\ t_j \le t_i + t_{ij} + (1 - x_{ij}^{k})\, T \end{cases} \qquad (i,j=1, \ldots, n;\ k=1, \ldots, \bar{m})$$

$$(68)\quad a_i \le t_i \le b_i \qquad (i=1, \ldots, n)$$

$$(69)\quad y_{ik} = 0, 1 \qquad (i=1, \ldots, n;\ k=1, \ldots, \bar{m})$$

$$(70)\quad x_{ij}^{k} = 0, 1 \qquad (i,j=1, \ldots, n;\ k=1, \ldots, \bar{m})$$

All constraints included in this formulation are self-explanatory or have been described previously.

Essentially, two well known problems are contained in (VRP3):

(i) the generalized assignment problem (GAP) (constraints (62), (63), (69));

(ii) the TSP: when the y_{ik}'s are fixed to satisfy the GAP constraints, then for a given k, constraints (64) - (66) and (70) define a TSP for vehicle k.

The authors propose an algorithm based on Benders' decomposition [2]. The method iterates between solving a GAP master problem to assign cities to vehicles and solving a TSP with time windows (TSPTW) to determine the best route for each vehicle. The method has the advantage of producing a feasible solution, even when not run to completion. Also, since it repeatedly solves a GAP and TSPTW, it can benefit directly from any improvement in algorithms for these two problems. Moreover, the method has an intuitive appeal in that its two phases correspond to the two steps naturally followed by vehicle dispatchers.

In [26], Fisher and Jaikumar use their algorithm to derive approximate solutions to a number of problems involving capacity restrictions only and ranging from 50 to 199 cities. The results indicate that their method compares favourably with those of Christofides et al. [8], Clarke and Wright [12] and Gillet and Miller [36]. It is worth mentioning that Toth [77] lately improved the two-phase method described in [8] to produce a method which outperforms in most instances that of Fisher and Jaikumar [26].

5.2 Two-index formulations

Two-index formulations can be derived from three-index formulations by aggregating all x_{ij}^k variables into a single variable x_{ij} indicating whether or not a vehicle travels directly from i to j in the optimal solution. In symmetrical problems, x_{ij} indicates the number of times (0, 1 or 2) edge (i,j) (i < j) is used in the optimal solution. More formally, x_{ij} is defined as

$$(71) \quad x_{ij} = \sum_{k=1}^{\bar{m}} x_{ij}^k \quad \text{or} \quad \sum_{k=1}^{m} x_{ij}^k$$

In two-index formulations, all vehicles are assumed to have the same costs and characteristics. These formulations were used for the exact solution of several types of VRPs by Laporte et al. [47-56] and Fleischmann [27,28]. The constraint relaxation algorithms developed by these authors belong to the same family as those used by several authors for the TSP and some of its extensions.

5.2.1 Symmetrical VRPs under capacity and distance restrictions

Two-index formulations have been mainly used for problems having a symmetrical distance matrix C. Consider first the single depot VRP un-

der capacity and distance restriction and assume C satisfies the triangle inequality. The following formulation is taken from [55]:

(VRP5) minimize $\sum_{i,j \in N} c_{ij} x_{ij}$

subject to

(72) $\sum_{j \in N-\{1\}} x_{1j} = 2m$

(73) $\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2$ $\qquad (k \in N-\{1\})$

(74) $\sum_{i,j \in S} x_{ij} \leq |S| - V(S)$ $\qquad (S \subseteq N-\{1\},\ |S| \geq 3)$

(75) $x_{ij} = \begin{cases} 0, 1, 2 \\ 0, 1 \end{cases}$ $\qquad \begin{matrix} (i=1,\ j \in N-\{1\}, \\ c_{1j} \leq \frac{1}{2}L) \end{matrix}$

(76) $m \geq 1$ and integer

In this formulation, L represents the maximum length of any route, m is the number of vehicles (a constant if it is fixed a priori or a variable) and V(S) represents a lower bound on the number of vehicles required to visit all cities of S in the optimal solution. Its computation is developed later. Variables are only defined if each of the following three conditions holds:

(i) $i < j$ (x_{ij} must be interpreted as x_{ji} whenever $i > j$);

(ii) $d_i + d_j < D$;

(iii) $c_{ij} + P(i) + P(j) \leq L$ where P(i) is the length of the shortest path from 1 to i. (When C satisfies the triangle inequality, P(i) is simply c_{1i}).

Constraints (72) and (73) specify the degree of each node while constraints (74) prohibit the formation of illegal subtours, i.e. subtours which are either

(i) disconnected from the depot, or

(ii) connected to the depot and having a total weight exceeding D, or

(iii) connected to the depot and having a total length exceeding L.

In [55], (VRP5) is solved by a constraint relaxation algorithm: integrality and subtour elimination constraints are first relaxed; integrality is obtained by branching on the variables while illegal subtours are eliminated as they are found to be violated (constraints (74) can be generated at non-integer solution). The algorithm can be summarized by the flow-chart depicted in figure 1 ($\bar{z}$ represents the objective value of the current subproblem while z^* is the value of the incumbent). Most steps of the algorithm require no explanation in so far as they correspond to a standard branch and bound procedure. However, it is worth examining two steps in some detail.

(i) Subtour prevention constraints

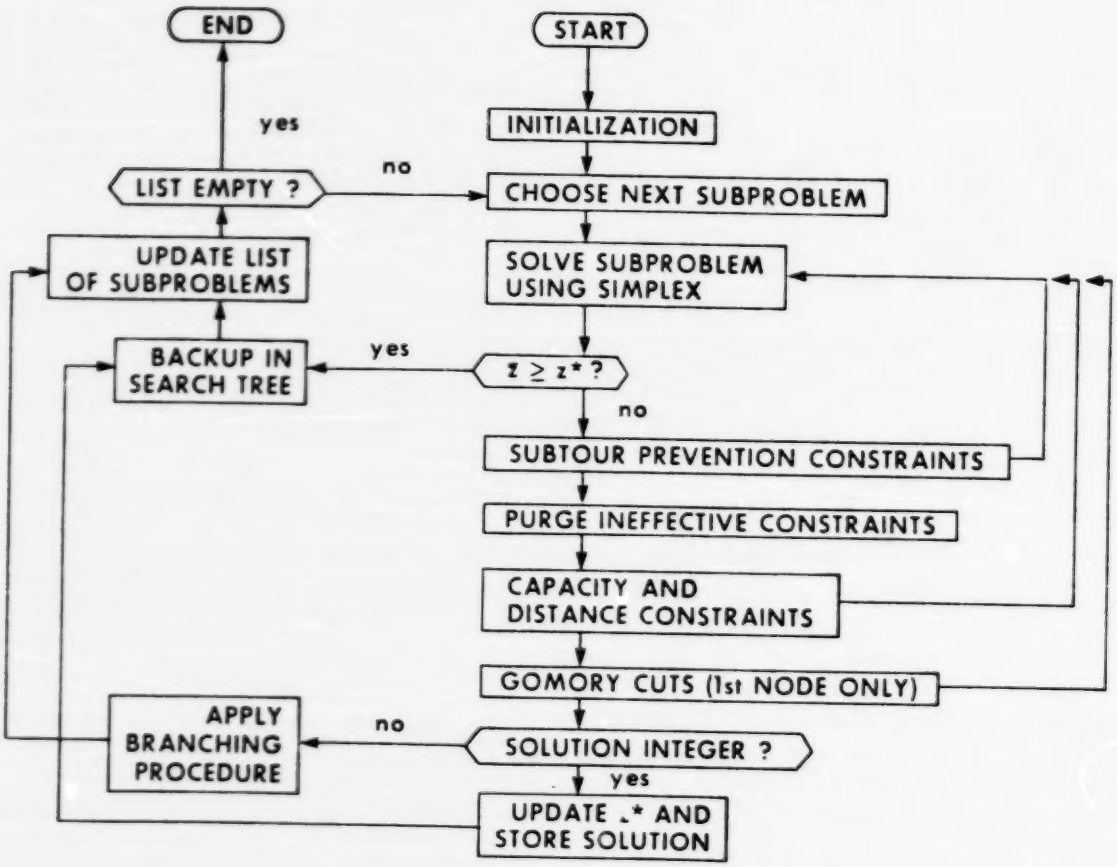
Consider the solution at a given node h of the search tree: this solution contains

(i) sets of nodes $\{i_1, \ldots, i_u\}$ $(u > 1)$ corresponding to chains $(i_1, \ldots, i_u)$ such that $1 \notin \{i_2, \ldots, i_{u-1}\}$ if $u > 2$ and for which *all* variables $x_{i_1 i_2}, x_{i_2 i_3}, \ldots, x_{i_{u-1} i_u}$ have been *fixed* at 1, and

(ii) nodes not belonging to such chains (we define for each such node i a set $\{i\}$).

FIGURE 1

Flow chart of the algorithm used
for the solution of (VRP5)



We refer to these sets of nodes S_k corresponding to chains or single nodes) as *components*. Each S_k has an associated weight $w(S_k)$ defined as

$$(77)\quad w(S_k) = \sum_{i \in S_k} d_i$$

and a length $\ell^h(S_k)$ defined as

$$(78)\quad \ell^h(S_k) = \begin{cases} \sum_{t=1}^{u-1} c_{i_t i_{t+1}} & \text{in the case of a chain } (i_1, \ldots, i_u) \\ 0 & \text{in the case of a node} \end{cases}$$

Consider a component S_r. If it corresponds to a chain, let p_r and q_r be the end nodes of that chain; if it corresponds to a node i, let $p_r = q_r = i$. In the first case, variable $x_{p_r q_r}$ can be *forced* to zero if

(i) $p_r, q_r \in N-\{1\}$ or

(ii) $p_r = 1$ and $\ell^h(S_r) + c_{1q_r} > L$.

Now consider two components S_r and S_s and let $i \in \{p_r,q_r\}$, $\{\bar{i}\} = \{p_r,q_r\} - \{i\}$, $j \in \{p_s,q_s\}$, $\{\bar{j}\} = \{p_s,q_s\} - \{j\}$. Variable x_{ij} can be forced to zero if

(iii) $i,j \in N-\{1\}$ and $w(S_r) + w(S_s) > D$, or

(iv) $i,j \in N-\{1\}$ and $P(\bar{i}) + P(\bar{j}) + \ell^h(S_r) + \ell^h(S_s) + c_{ij} > L$, or

(v) $i=1$, $j \in N-\{1\}$ and $P(\bar{j}) + \ell^h(S_s) + c_{1j} > L$.

At node h of the search tree, consider the graph $G' = (N-\{1\},E',C)$ where E' is the set of all (undirected) edges (i,j) for which $i,j \in N-\{1\}$ and $x_{ij} > 0$. $N-\{1\}$ can be partitioned into K *connected components* $S_1, S_2, \ldots, S_K$ i.e. sets of nodes linked by "positive edges". Let $S_k^+ = S_k \cup \{1\}$. Each S_k^+ has a weight

$$(79)\quad w(S_k^+) = w(S_k) = \sum_{i \in S_k} d_i$$

and a length

$$(80)\quad \ell^h(S_k^+) = \sum_{i,j \in S_k^+} c_{ij}\, x_{ij}$$

It can be shown that the minimum number of vehicles required to visit all nodes of S_k^+ in the optimal solution is bounded below by

$$(81)\quad V^h(S_k) = \max \{\lceil w(S_k)/D \rceil,\ \lceil \ell^h(S_k^+)/L \rceil\}.$$

Therefore, for each k, we can impose the following subtour elimination constraint, whenever it is violated.

$$(82)\quad \sum_{i,j \in S_k} x_{ij} \leq |S_k| - V^h(S_k)$$

Recently, Gavish [33] suggested the use of a sharper bound $V_1^h(S_k)$ obtained by replacing in (81) $\lceil w(S_k)/D \rceil$ by the number of bins in the optimal solution of the bin packing problem [23] associated with S_k. Alternatively, one could use one of the known lower bounds for the bin packing problem [61].

In addition to imposing one type (82) constraint for each S_k, the authors also introduce a similar constraint for $S^* = S_1 \cup S_2 \cup \ldots \cup S_K$, whenever $K > 1$; this constraint is obtained by replacing S_k by S^* in (81) and (82). Computational experiments show that the presence of such a constraint helps to reduce the growth of the search tree.

Since constraints (82) are sometimes imposed when the current solution is fractional (see Figure 1), they may be satisfied at the time of their generation. In such cases, a violated constraint may be derived by means of a simple heuristic search [55].

It can be shown that if C does not satisfy the triangle inequality, $\lceil \gamma^h(S_k^+)/L \rceil$ does not constitute a valid lower bound on the number of vehicles required to visit all nodes of S_k' in the optimal solution of subproblem h. In such cases, a valid constraint can be derived to eliminate subtours connected to the depot and which do not satisfy distance constraints. This constraint, which is derived in [55], can be expressed as

$$(83) \quad \sum_{j \in S-\{1\}} x_{1j} + 3 \sum_{\substack{i \in S-\{1\}, j \in \bar{S} \\ \text{or } i \in \bar{S}, j \in S-\{1\}}} x_{ij} \geq 4$$

In (83), S is a subset of N satisfying

(i) $1 \in S$;

(ii) $|S| \geq 3$;

(iii) the value of the TSP solution on S exceeds L. (This information is available when the constraint is generated.)

Using the algorithm summarized by Figure 1 (or a slightly modified version for problems in which C did not satisfy the triangle inequality), Laporte, Nobert and Desrochers [55] have obtained exact solutions to problems involving up to 50 cities for problems in which C satisfied the triangle inequality and up to 60 cities for problems in which C did not possess this property.

(VRP5) can be stengthened by the inclusion of extra constraints such as *comb inequalities*. These constraints were first developed by Chvátal [11] for the TSP and later generalized by Grötschel and Padberg [40]. They constitute facets of the integer polytope associated with the TSP [41] and include subtour elimination constraints (6) as a special case. Their inclusion in a constraint relaxation algorithm for the TSP enabled Crowder and Padberg to solve exactly a 318 city problem [13].

Comb inequalities for the TSP were generalized for the VRP by Laporte and Nobert [53] and later modified by Laporte and Bourjolly [48]. In the case of the CVRP, they can be stated as follows. Let $W_\ell (\ell=0, \ldots, k)$ be subsets of N-{1} satisfying

$$(84) \quad |W_\ell - W_0| \geq 1 \qquad (\ell=1, \ldots, k)$$

$$(85) \quad |W_\ell \cap W_0| \geq 1 \qquad (\ell=1, \ldots, k)$$

$$(86) \quad |W_\ell \cap W_{\ell'}| = 0 \qquad (1 \leq \ell < \ell' \leq k)$$

Then the following condition (comb inequality) holds for every feasible solution of the CVRP:

$$(87) \quad \sum_{\ell=0}^{k} \sum_{i,j \in W_\ell} x_{ij} \leq |W_0| + \sum_{\ell=1}^{k} |W_\ell| - \left\lceil \frac{1}{2} \sum_{\ell=1}^{k} [V_1(W_\ell)+V_1(W_\ell - W_0)+V_1(W_\ell \cap W_0)] \right\rceil$$

where $V_1(S)$ is a lower bound on the optimal value of the bin packing solution associated with S.

It can be shown [48] that in general, comb inequalities defined by (87) do not constitute facets of the integer polytope associated with the CVRP.

5.2.2 Capacitated symmetrical road-TSP

Fleischmann [27,28] uses the expression "road-TSP" to designate routing problems derived from a real-life (sparse) road network and in which multiple passages through the same city are allowed whenever they are economical. Such problems have also been referred to as complete cycle problems by other authors [65]. The two-index formulation used by Fleischmann belongs to the same family as those of Dantzig et al. [14],

Miliotis [62-64], Land [46] and Crowder and Padberg [13]. It includes degree constraints, integrality constraints, connectivity constraints (preventing the occurence of disconnected components) and a new class of constraints called *3-star constraints*, generalizing comb inequalities. Fleischmann uses a pure cutting planes method and obtains exact solutions for road-TSP's involving up to 292 cities [28]. Fleischmann applies a similar approach to the capacitated road-TSP (CRTSP) [27]. He first considers the following relaxation of the problem:

(VRP6) minimize $\sum_{i,j \in N} c_{ij} x_{ij}$

subject to

(88) $\sum_{j=2}^{m} x_{1j} \geq 2m$

(89) $\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} \geq 2$ and even $\qquad (k \in N-\{1\})$

(90) $\sum_{\substack{i \in S, j \in \bar{S} \\ \text{or } i \in \bar{S}, j \in S}} x_{ij} \geq 2\, V(S) \qquad (2 \leq |S| \leq n-2;\ S \subset N-\{1\})$

(91) $x_{ij} \geq 0$ and integer $\qquad (i < j)$

(92) $m \geq 1$ and integer

Here, V(S) is defined as $\left\lceil \sum_{i \in S} d_i / D \right\rceil$, but could easily be replaced by the bin packing bound. (VRP6) is solved by means of a constraint relaxation algorithm of the type used for the road-TSP. 3-star constraints generalizing those used for the road-TSP (in the same way as comb inequalities used for the CVRP [53] generalize those of Grötschel and Padberg [40,41]), are generated to eliminate fractional solutions not violating connectivity constraints (90). If the solution to (VRP6) is feasible for

the CRTSP, it is then optimal; otherwise, it constitutes a valid lower bound on the value of the CRTSP solution. Fleischmann [27] gives an example of a solution satisfying (89) - (92), but infeasible for the CRTSP. Laporte [47] describes a procedure for generating constraints in order to eliminate such solutions.

Using this approach, Fleischmann succeeded in solving to optimality some CRTSPs ranging from 28 to 68 cities [27].

5.2.3 Capacitated location-routing symmetrical problems

Laporte, Nobert and Arpin [54] used a two-index formulation to treat a family of CVRPs involving simultaneous depot location and routing. Consider $R \subseteq N$, a set of *potential depots*. The number P of such nodes used as depots in the optimal solution must lie between two prespecified bounds $\underline{P} \geq 1$ and $\bar{P} \leq |R|$. The cost of using node r as a depot is equal to g_r. There are m_r identical vehicles based at depot r, each with the same capacity D and a fixed cost f_r. To each node i of N-R, is associated a nonnegative requirement d_i ($\leq D$).

The problem consists of selecting depot sites (when $\underline{P} < |R|$), of determining how many vehicles are based at each selected depot and of establishing vehicle routes in such a way that

(i) each route starts and ends at the same depot;

(ii) all requirements are met exactly once by a vehicle (the same city i may be visited more than once if this saves distance but then, the requirement d_i of that city is satisfied only once by one vehicle);

(iii) the sum of all requirements satisfied by any vehicle does not exceed D;

(iv) $\underline{P} \leq P \leq \bar{P}$;

(v) for each node r used as a depot, the number of vehicles lies between two prespecified bounds $\underline{m}_r$ and $\bar{m}_r$: $1 \leq \underline{m}_r \leq m_r \leq \bar{m}_r$;

(vi) the total cost is minimized.

In addition to the notation already introduced, define

T : an arbitrarily large number

x_{ij} : a variable indicating the number of times edge (i,j) is used in the optimal solution. x_{ij} is not defined if $i \geq j$, if $i,j \in R$ or if $d_i + d_j > D$. x_{ij} must be interpreted as x_{ji} whenever $i > j$

y_r : a binary variable indicating whether node r is used as a depot ($y_r = 1$) or not ($y_r = 0$)

The problem can be formulated as follows:

(VRP7) minimize $\sum_{i,j \in N} c_{ij} x_{ij} + \sum_{r \in R} (g_r y_r + f_r m_r)$

subject to

(93) $\sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2$ $\qquad (k \in N\text{-}R)$

(94) $\sum_{i<r} x_{ir} + \sum_{r<j} x_{rj} = 2m_r$ $\qquad (r \in R)$

(95) $\sum_{i,j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{k \in S} d_k}{D} \right\rceil$ $\qquad (S \subseteq N\text{-}R,\ |S| \geq 3)$

(96) $x_{i_1 i_2} + 3x_{i_2 i_3} + x_{i_3 i_4} \leq 4$ $\qquad (i_1, i_4 \in R;\ i_2, i_3 \in N\text{-}R)$

(97) $x_{i_1 i_2} + x_{i_{h-1} i_h} + 2 \sum_{i,j \in \{i_2, \ldots, i_{h-1}\}} x_{ij} \leq 2h-5$ $\qquad (h \geq 5;\ i_1, i_h \in R;\ i_2, \ldots, i_{h-1} \in N\text{-}R;\ d_{i_2} + \ldots + d_{i_{h-1}} \leq D)$

$$(98)\quad y_r \le m_r \le T y_r \qquad (r \in R)$$

$$(99)\quad \underline{m}_r \le m_r \le \bar{m}_r \qquad (r \in R)$$

$$(100)\quad \underline{P} \le \sum_{r \in R} y_r \le \bar{P}$$

$$(101)\quad y_r = 0, 1 \qquad (r \in R)$$

$$(102)\quad x_{ij} = \begin{cases} 0, 1 & (i,j \in N\text{-}R) \\ 0, 1, 2 & (i \text{ or } j \in R) \end{cases}$$

In this formulation, constraints (93) specify that each city not used as a depot must be serviced exactly once by a vehicle. Similarly, constraints (94) express the fact that m_r vehicles must leave and enter each city in R. Constraints (95) ensure that the solution does not contain any illegal subtours, i.e. subtours disjoint from R or subtours having a total weight exceeding D. As previously, the bin packing bound could be substituted in the right hand side of (95). Constraints (96) and (97) are *chain barring constraints*. They ensure that each route starts and ends at the same depot. Their development is fairly lengthy and is fully provided in [54].

Using a constraint relaxation approach in which integrality constraints, subtour elimination constraints and chain barring constraints are first relaxed, the authors succeeded in solving to optimality

(i) problems without capacity restrictions involving up to 25 cities;

(ii) problems with capacity restrictions containing up to 20 cities.

It was observed that imposing large depot costs tended to produce easier problem (the locational aspect of the problem becoming less predominent); on the other hand, the size of vehicle costs did not seem to affect the behaviour of the algorithm one way or the other.

Other types of less sophisticated location-routing problems are also treated in [51,56].

5.2.4 Asymmetrical VRPs under capacity or distance restrictions

Many of the concepts developed for symmetrical VRPs are also usable in the asymmetrical case. Two types of asymmetrical VRPs (the CVRP and the DVRP) were studied in [49] and [50]. It was assumed in each case that an upper bound $\bar{m}$ on the number of vehicles was known. Then the graph $G = (N,E,C)$ associated with the VRP was extended into a graph $G' = (N',E',C')$ as suggested by Lenstra and Rinnooy Kan [57] for example. First, $\bar{m}$ artificial depots were introduced; let $N^* = \{1, n+1, n+2, \ldots, n+\bar{m}-1\}$ be the set of depots and let $N' = N \cup N^*$. Define $C' = (c'_{ij})$ as follows:

$$(103)\quad c'_{ij} = \begin{cases} c_{ij} & (i,j \in N,\ i \neq j) \\ c_{i1} & (i \in N,\ j \in N^*) \\ c_{1j} & (i \in N^*,\ j \in N) \\ \lambda & (i,j \in N^*,\ i \neq j) \\ \infty & (i = j) \end{cases}$$

As explained in section 2, the nature of the problem varies according to λ.

The problem is formulated as (TSP1) (see section 2.1) where N is replaced by N'. The solution of the TSP on G' can easily be transformed into the solution of a multiple TSP on G where $m \leq \bar{m}$:

(i) if $i \in N-\{1\}$ and $j \in N'-N$, replace (i,j) by $(i,1)$;

(ii) if $i \in N'-N$ and $j \in N-\{1\}$, replace (i,j) by $(1,j)$;

(iii) if $i,j \in N'-N$, remove (i,j).

The transformed solution consists of at most $\bar{m}$ Hamiltonian circuits covering all nodes of G and having a unique common node at 1. This solution is feasible for the VRP if and only if

(i) the weight of each Hamiltonian circuit is less than or equal to D (in the CVRP);

(ii) the length of each Hamiltonian circuit is less than or equal to L (in the DVRP).

These two types of infeasibilities can be eliminated at the source, during the solution of the TSP of G'. This is done by extending or otherwise modifying the Carpaneto and Toth algorithm [5] for asymmetrical TSPs. Using this approach, exact solutions were obtained for problems containing up to 260 cities in the case of the CVRP and up to 400 cities in the case of the DVRP.

6. COMMODITY FLOW FORMULATIONS

In addition to the x_{ij}^k (or x_{ij}) variables used in the CVRP formulations of the previous section, commodity flow formulations associate flow variables y_{ij}^ℓ (or y_{ij}) with the arcs. These variables indicate how much of the demand destined for city ℓ travels on arc (i,j). Such formulations were first proposed by Garvin et al. [31] in an oil delivery problem. Many variants were later developed and analyzed by Gavish et al. [34,35] for various TSP extensions. As far as the authors are aware, no exact VRP algorithm based on a vehicle flow formulation has ever been constructed and tested. This section presents some of the commodity flow formulations set up for the CVRP. They are adapted from [33,35].

First consider CVRPs in which d_i represents the *demand* of city i and where the number m of vehicles is fixed. The following formulation is adapted from the work of Gavish et al. [31] (these authors also consider time limit constraints on the routes).

$$\text{(VRP8)} \quad \text{minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\text{(104)} \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad (j=2, \ldots, n)$$

$$\text{(105)} \quad \sum_{j=1}^{n} x_{ij} = 1 \qquad (i=2, \ldots, n)$$

$$\text{(106)} \quad \sum_{i=2}^{n} x_{i1} = \sum_{j=2}^{n} x_{1j} = m$$

$$\text{(107)} \quad \sum_{i=1}^{n} y_{ij}^\ell - \sum_{i=1}^{n} y_{ji}^\ell = \begin{cases} d_\ell & (j=\ell;\ j=2, \ldots, n) \\ 0 & (j\neq\ell;\ j,\ell=2, \ldots, n) \end{cases}$$

$$(108)\quad \sum_{\ell=1}^{n} y_{ij}^{\ell} \leq D\, x_{ij} \qquad (i\neq j;\ i,j= , \ldots, n)$$

$$(109)\quad y_{ij}^{\ell} \geq 0 \qquad (i,j,\ell=1, \ldots, n)$$

$$(110)\quad x_{ij} = 0, 1 \qquad (i,j=1, \ldots, n)$$

In this formulation, constraints (104), (105) and (110) ensure that every city, except the depot is visited exactly once; constraints (106) and (110) specify that there must be m vehicles; constraints (107) are flow balancing constraints which ensure that the demand of each node is satisfied; constraints (107) and (108) ensure that the solution will contain no illegal subtours; constraints (108) are imposed in order to ensure that vehicle capacities are never exceeded and that goods can only be moved through an arc if a vehicle uses this arc.

In the case where the number of vehicles is bounded by $\bar{m}$, constraints (106) can be replaced by

$$(106')\quad \sum_{i=2}^{n} x_{i1} = \sum_{j=2}^{n} x_{1j} \leq \bar{m}$$

Similarly, if the number of vehicles is unspecified, constraints (106) become

$$(106'')\quad \sum_{i=2}^{n} x_{i1} = \sum_{j=2}^{n} x_{1j}$$

As suggested by Magnanti [60], such a model can easily be transformed to deal with the case of heterogeneous fleets. This is done by defining variables x_{ij}^{k} indicating whether or not vehicle k travels directly from i to j. Constraints involving x_{ij} variables in (VRP8) must of course be modified accordingly.

In Gavish and Graves [34], the index ℓ is removed and the problem becomes:

$$\text{(VRP9)} \quad \text{minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to (104) - (106), (110) and

$$\text{(111)} \quad \sum_{j=1}^{n} y_{ij} - \sum_{j=1}^{n} y_{ji} = d_i \qquad (i=2, \ldots, n)$$

$$\text{(112)} \quad y_{ij} \leq D\, x_{ij} \qquad (i,j=1, \ldots, n)$$

$$\text{(113)} \quad y_{ij} \geq 0 \qquad (i,j=1, \ldots, n)$$

In (VRP9), y_{ij} represents the quantity of goods travelling on arc (i,j).

Using different arguments, Magnanti [60] and Nobert [68] have proved that by applying Benders' decomposition to (VRP8) or (VRP9), one obtains the subtour elimination constraints vehicle flow model (VRP5) specialized to the CVRP.

In [33], Gavish presents some more constrained formulations.

$$\text{(VRP10)} \quad \text{minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to (104) - (106), (109), (110) and

$$\text{(114)} \quad \sum_{i=1}^{n} y_{ij}^{\ell} - \sum_{i=1}^{n} y_{ji}^{\ell} = \begin{cases} d_\ell & (j=\ell;\ \ell=2, \ldots, n) \\ 0 & (j\neq\ell;\ j,\ell=2, \ldots, n) \\ -d_\ell & (j=1;\ \ell=2, \ldots, n) \end{cases}$$

$$(115)\quad y_{ij}^{\ell} \le d_{\ell}\, x_{ij} \qquad (i,\ell=2, \ldots, n;\ j=1, \ldots, n)$$

$$(116)\quad \sum_{j=2}^{n} \sum_{\ell=2}^{n} y_{ji}^{\ell} \le D - d_i \qquad (i=1, \ldots, n;\ d_1 = 0)$$

The elimination of index ℓ yields a simpler model:

$$(\text{VRP11})\quad \text{minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

subject to (104) - (106), (110), (111), (113) and

$$(117)\quad d_i\, x_{ij} \le y_{ij} \le (D - d_i)\, x_{ij} \qquad (i,j=1, \ldots, n)$$

In order to establish comparisons between these formulations, consider LPi, the value of the LP relaxation of (VRPi). Gavish [33] shows that

(i) LP10 $\ge$ LP11 (This is a generalization of a similar result for the TSP proved in [79]

(ii) LP5 = LP11 (when $V(S) = \sum_{i \in S} d_i/D$ in (74))

(iii) LP5 $\ge$ LP10 $\ge$ LP11 (when V(S) = bin packing bound in (74))

These relationships and the equivalence between (VRP8) and (VRP5) suggest that it may be relatively efficient to use (VRP5) with the bin packing bound to solve the CVRP.

In [34], Gavish and Graves report some computational results relative to the bound associated with various relaxations of (VRP11) for CVRPs ranging from 10 to 30 cities. These results show the presence of fairly large gaps between the VRP optimum and the LP optimum when $n \ge 15$. Finally, we mention the existence of various commodity flow formulations for other types of VRPs. These were developed by Gavish and Graves [34,35] for multi-depot VRPs, dial-a-ride problems and school bus problems.

7. CONCLUSION

Many person-years of research time have been spent on the development of solution methods for the VRP. Yet, despite all these efforts, most types of VRPs remain virtually unsolved: exact methods can only handle problems of relatively modest dimensions.

Of all methods considered in this survey, direct search algorithms offer, in our opinion, the least potential for growth. Dynamic programming algorithms, on the other hand do not appear to have reached their maximum potential; the state-space relaxation approach developed in [10] does not seem to have been exploited fully. Similarly, several ILP formulations have been provided, but few have led to the construction of exact algorithms. The rare cases that have been published would indicate that ILP works quite efficiently on some types of problems.

The most successful algorithms presented in this survey take the greatest possible advantage of the problem on hand. Tight problems can not be handled in the same manner as loose problems and, in general, symmetrical and asymmetrical problems require different types of algorithms. One obvious tendency has been to move from general to particularized algorithms.

Looking ahead, it is safe to say that the study and development of exact algorithms for the vehicle routing problem will remain a major research topic in years to come and should yield new theoretical developments.

BIBLIOGRAPHY

[1] BALINSKI, M., QUANDT, R.: "On an Integer Program for a Delivery Problem", Operations Research, vol.12, pp.300-304, 1964.

[2] BENDERS, J.F.: "Partitioning Procedures for Solving Mixed-Variables Programming Problems", Numerische Mathematik, vol.4, pp.238-252, 1962.

[3] BODIN, L.D., GOLDEN, B.L.: "Classification in Vehicle Routing and Scheduling", Networks, vol.11, pp.97-108, 1981.

[4] BODIN, L.D., GOLDEN, B.L., ASSAD, A., BALL, M.: "Routing and Scheduling of Vehicles and Crews, The State of the Art", Computers & Operations Research, vol.10, pp.69-211, 1983.

[5] CARPANETO, G., TOTH, P.: "Some New Branching and Bounding Criteria for the Asymmetric Travelling Salesman Problem", Management Science, vol.26, pp.736-743, 1980.

[6] CHRISTOFIDES, N.: "The Vehicle Routing Problem", RAIRO (recherche opérationnelle), vol.10, pp.55-70, 1976.

[7] CHRISTOFIDES, N., EILON, S.: "An Algorithm for the Vehicle Dispatching Problem", Operational Research Quarterly, vol.20, pp.309-318, 1969.

[8] CHRISTOFIDES, N., MINGOZZI, A., TOTH, P.: "The Vehicle Routing Problem", Combinatorial Optimization, pp.315-338, Wiley, 1979.

[9] CHRISTOFIDES, N., MINGOZZI, A., TOTH, P.: "Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations", Mathematical Programming, vol.20, pp.255-282, 1981.

[10] CHRISTOFIDES, N., MINGOZZI, A., TOTH, P.: "Space State Relaxation Procedures for the Computation of Bounds to Routing Problems", Networks, vol.11, pp.145-164, 1981.

[11] CHVÁTAL, V.: "Edmonds Polytopes and Weakly Hamiltonian Graphs", Mathematical Programming, vol.5, pp.29-40, 1973.

[12] CLARKE, G., WRIGHT, J.W.: "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", Operations Research, vol.12, pp.568-581, 1964.

[13] CROWDER, H., PADBERG, M.W.: "Solving Large-Scale Symmetric Travelling Salesman Problems to Optimality", Management Science, vol.26, pp.495-509, 1980.

[14] DANTZIG, G.B., FULKERSON, D.R., JOHNSON, S.M.: "Solution of a Large Scale Travelling Salesman Problem", Operations Research, vol.2, pp.393-410, 1954.

[15] DANTZIG, G.B., RAMSER, J.: "The Truck Dispatching Problem", Management Science, vol.6, pp.81-91, 1959.

[16] DESROCHERS, M., SOUMIS, F.: "A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows", Publication 394A, Centre de Recherche sur les Transports, Université de Montréal, 27 pp., 1985.

[17] DESROSIERS, J., DUMAS, Y., SOUMIS, F.: "The Multiple Vehicles Many to Many Routing Problem with Time Windows", Cahiers du GERAD, G-84-13, Ecole des Hautes Etudes Commerciales de Montréal, 24 pp., 1984.

[18] DESROSIERS, J., DUMAS, Y., SOUMIS, F.: "A Dynamic Programming Method for the Large Scale Single Vehicle Dial-a-Ride with Time Windows", Publication 361, Centre de Recherche sur les Transports, Université de Montréal, 24 pp., 1984. Forthcoming in American Journal of Mathematical and Management Sciences.

[19] DESROSIERS, J., PELLETIER, P., SOUMIS, F.: "Plus court chemin avec contraintes d'horaires", RAIRO (Recherche opérationnelle), vol.17, pp.357-377, 1983.

[20] DESROSIERS, J., SAUVE, M., SOUMIS, F.: "Shortest Path with Time Windows Problem", Publication 389, Centre de Recherche sur les Transports, Université de Montréal, 13 pp., 1985.

[21] DESROSIERS, J., SOUMIS, F., DESROCHERS, M.: "Routing with Time Windows by Column Generation", Networks, vol.14, pp.545-565, 1984.

[22] DIJKSTRA, E.W.: "A Note on Two Problems in Connexion with Graphs", Numerische Mathematik, vol.1, pp.269-271, 1959.

[23] EILON, S., CHRISTOFIDES, N.: "The Loading Problem", Management Sciences, vol.17, pp.259-267, 1971.

[24] EILON, S., WATSON-GANDY, C.D.T., CHRISTOFIDES, N.: "Distribution Management: Mathematical Modelling and Practical Analysis", Griffin, 238 pp., 1971.

[25] FISHER, M., JAIKUMAR, R.: "A Decomposition Algorithm for Large-Scale Vehicle Routing", Working paper 78-11-05, Department of Decision Sciences, University of Pennsylvania, 27 pp., 1978.

[26] FISHER, M., JAIKUMAR, R.: "A Generalized Assignment Heuristic for Vehicle Routing", Networks, vol.11, pp.109-124, 1981.

[27] FLEISCHMANN, B.: "Linear Programming Approaches to Travelling Salesman and Vehicle Scheduling Problems", Paper presented at the XI. International Symposium on Mathematical Programming, 20 pp., Bonn, August 1982.

[28] FLEISCHMANN, B.: "A Cutting Planes Procedure for the Travelling Salesman Problem on Road Networks", Research Report QM-02-83, University of Hamburg, 21 pp., 1983.

[29] FORD, L.R., FULKERSON, D.R.: "Flows in Networks", Princeton University Press, 194 pp., 1962.

[30] FOSTER, B.A., RYAN, D.M.: "An Integer Programming Approach to the Vehicle Scheduling Problem", Operational Research Quarterly, vol.27, pp.367-384, 1976.

[31] GARVIN, W.M., CRANDALL, H.W., JOHN, J.B., SPELLMAN, R.A.: "Applications of Linear Programming in the Oil Industry", Management Science, vol.3, pp.407-430, 1957.

[32] GASKELL, T.J.: "Bases for Vehicle Fleet Scheduling", Operational Research Quarterly, vol.18, pp.281-295, 1967.

[33] GAVISH, B.: "The Delivery Problem: New Cutting Planes Procedures", paper presented at the TIMS XXVI Conference, Copenhagen, June 1984.

[34] GAVISH, B., GRAVES, S.C.: "The Travelling Salesman Problem and Related Problems", Working Paper no.7905, Graduate School of Management, University of Rochester, 32 pp., 1979.

[35] GAVISH, B., GRAVES, S.C.: "Scheduling and Routing in Transportation and Distribution Systems: Formulation and New Formulations", Working Paper, Graduate School of Management, University of Rochester, 35 pp., 1982.

[36] GILLET, B.E., MILLER, L.R.: "A Heuristic Algorithm for the Vehicle-Dispatch Problem", Operations Research, vol.22, pp.341-349, 1974.

[37] GOLDEN, B., BODIN, L., DOYLE, T., STEWART, W.: "Approximate Travelling Salesman Algorithms", Operations Research, vol.28, pp.694-711, 1980.

[38] GOLDEN, B., MAGNANTI, T.L., NGUYEN, H.Q.: "Implementing Vehicle Routing Algorithms", Networks, vol.7, pp.113-148, 1977.

[39] GOLDEN, B., STEWART, W.: "The Empirical Analysis of TSP Heuristics", 85 pp., 1981 (Forthcoming in "The Travelling Salesman Problem", E. Lawler et al. eds.).

[40] GRÖTSCHEL, M., PADBERG, M.W.: "On the Symmetric Travelling Salesman Problem I: Inequalities", Mathematical Programming, vol.16, pp.265-280, 1979.

[41] GRÖTSCHEL, M., PADBERG, M.W.: "On the Symmetric Travelling Salesman Problem II: Lifting Theorems and Facets", Mathematical Programming, vol.16, pp.281-302, 1979.

[42] HAYS, R.: "The Delivery Problem", Management Science Research Report no.106, Carnegie Institute of Technology, 1967.

[43] HELD, M., KARP, R.M.: "A Dynamic Programming Approach to Sequencing Problems", SIAM 10, pp.196-210, 1962.

[44] HOUCK, D., PICARD, J., QUEYRANNE, M., VEMUGANTI, R.: "The Travelling Salesman Problem as a Constrained Shortest Path Problem: Theory and Computational Experience", Opsearch, vol.17, pp.93-109, 1980.

[45] KRUSKAL, J.B.: "On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem", Proceedings of the American Mathematical Society, vol.2, pp.48-50, 1956.

[46] LAND, A.H.: "The Solution of Some 100-City Travelling Salesman Problems", paper presented at the Tenth International Symposium on Mathematical Programming, Montreal, 1979.

[47] LAPORTE, G.: "An Integer Programming Approach to the Vehicle Scheduling Problem", Cahiers du GERAD, G-82-10, Ecole des Hautes Etudes Commerciales de Montréal, 9 pp., 1982.

[48] LAPORTE, G., BOURJOLLY, J.-M.: "Some Further Results on k-Star Constraints and Comb Inequalities", Cahiers du GERAD, G-84-17, Ecole des Hautes Etudes Commerciales de Montréal, 9 pp., 1984.

[49] LAPORTE, G., MERCURE, H., NOBERT. Y.: "An Exact Algorithm for the Asymmetrical Capacitated Vehicle Routing Problem", Forthcoming in Networks, 1986.

[50] LAPORTE, G., NGUYEN, T., NOBERT, Y.: "A Branch and Bound Algorithm for the Asymmetrical Distance Constrained Vehicle Routing Problem", Cahiers du GERAD, G-85-05, Ecole des Hautes Etudes Commerciales de Montréal, 18 pp., 1985.

[51] LAPORTE, G., NOBERT, Y.: "An Exact Algorithm for Minimizing Routing and Operating Costs in Depot Location", European Journal of Operational Research, vol.6, no 2, pp.224-226, 1981.

[52] LAPORTE, G., NOBERT, Y.: "Algorithmes de relaxation de contraintes pour le problème du voyageur de commerce symétrique et ses extensions", Annales des Sciences mathématiques du Québec, vol.VII, pp.109-137, 1983.

[53] LAPORTE, G., NOBERT, Y.: "Comb Inequalities for the Vehicle Routing Problem", Methods of Operations Research, vol.51, pp.271-276, 1984.

[54] LAPORTE, G., NOBERT, Y., ARPIN, D.: "An Exact Algorithm for Solving a Capacitated Location-Routing Problem", Forthcoming in Annals of Operations Research, 1986.

[55] LAPORTE, G., NOBERT, Y., DESROCHERS, M.: "Optimal Routing Under Capacity and Distance Restrictions", Forthcoming in Operations Research, 1985.

[56] LAPORTE, G., NOBERT, Y., PELLETIER, P.: "Hamiltonian Location Problems", European Journal of Operational Research, vol.12, no 1, pp.82-89, 1983.

[57] LENSTRA, J.K., RINNOOY KAN, A.H.G.: "Some Simple Applications of the Travelling Salesman Problem", Operational Research Quarterly, vol.26, pp.717-734, 1975.

[58] LENSTRA, J.K., RINNOOY KAN, A.H.G.: "Complexity of Vehicle Routing and Scheduling Problems", Networks, vol.11, pp.221-228, 1981.

[59] LITTLE, D.C., MURTY, K.G., SWEENEY, D.W., KAREL, C.: "An Algorithm for the Travelling Salesman Problem", Operations Research, vol.11, pp.972-989, 1963.

[60] MAGNANTI, T.L.: "Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects", Networks, vol.11, pp.179-214, 1981.

[61] MARTELLO, S., TOTH, P.: "Lower Bounds and Reduction Procedures for the Bin Packing Problem", Working Paper 1-84, Progretto e Analisi di Algoritmi, University of Bologna, 14 pp., 1984.

[62] MILIOTIS, P.: "Combining Cutting Planes and Branch-and-Bound Methods to Solve Integer Programming Problems: Applications to the Travelling Salesman Problem and the 1-Matching Problems", Ph.D. Thesis, University of London, 1975.

[63] MILIOTIS, P.: "Integer Programming Approaches to the Travelling Salesman Problem", Mathematical Programming, vol.10, pp.367-378, 1976.

[64] MILIOTIS, P.: "Using Cutting Planes to Solve the Symmetric Travelling Salesman Problem", Mathematical Programming, vol.15, pp.177-188, 1978.

[65] MILIOTIS, P., LAPORTE, G., NOBERT, Y.: "Computational Comparison of Two Methods for Finding the Shortest Complete Cycle or Circuit in a Graph", RAIRO (recherche opérationnelle), vol.15, pp.233-239, 1981.

[66] MILLER, C.E., TUCKER, A.W., ZEMLIN, R.A.: "Integer Programming Formulation of Travelling Salesman Problems", JACM, vol.7, pp.326-329, 1969.

[67] MOLE, R.: "A Survey of Local Delivery Vehicle Routing Methodology", Journal of the Operational Research Society, vol.30, pp.245-252, 1979.

[68] NOBERT, Y.: "Construction d'algorithmes optimaux pour des extensions au problème du voyageur de commerce", Thèse de doctorat, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, 240 pp., 1982.

[69] ORLOFF, C.: "Route Constrained Fleet Scheduling", Transportation Science, vol.10, pp.149-168, 1976.

[70] PIERCE, J.F.: "Direct Search Algorithms for the Truck-Dispatching Problem", Transportation Research, vol.3, pp.1-42, 1969.

[71] PSARAFTIS, H.N.: "A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem", Transportation Science, vol.14, pp.130-154, 1980.

[72] PSARAFTIS, H.N.: "An Exact Algorithm for the Single Vehicle Many-to-Many Dial-a-Ride Problem with Time Windows", Transportation Science, vol.17, pp.351-360, 1983.

[73] PSARAFTIS, H.N.: "k-Interchange Procedures for Local Search in a Precedence Constrained Routing Problem", European Journal of Operational Research, vol.13, pp.391-402, 1983.

[74] RAO, M.R., ZIONTS, S.: "Allocation of Transportation Units to Alternative Trips - A Column Generation Scheme with Out-of-Kilter Subproblems", Operations Research, vol.16, pp.52-63, 1968.

[75] SIMPSON, R.W., KIVESTU, P.: "Network Models in Transportation: a Bibliography", Report 76-11, Center for Transportation Studies, Massachusetts Institute of Technology, 40 pp., 1976.

[76] TOREGAS, C., REVELLE, C.: "Location Under Time or Distance Constraints", Papers of the Regional Science Association, vol.28, pp.133-143, 1972.

[77] TOTH, P.: "Heuristic Algorithms for the Vehicle Routing Problem", paper presented at the Workshop on Routing Problems, Hamburg, June 1984.

[78] WATSON-GANDY, C.D.T., FOULDS, L.R.: "The Vehicle Scheduling Problem: a Survey", New Zealand Operational Research, vol.9, pp.73-92, 1981.

[79] WONG, R.T.: "Integer Programming Formulations of the Travelling Salesman Problem", Proceedings of the IEEE International Conference on Circuits and Computers, pp.149-152, 1980.

[80] WREN, A., HOLLIDAY, A.: "Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points", Operational Research Quarterly, vol.23, pp.333-344, 1972.